

REMARKS

The Office Action mailed June 7, 2006, considered claims 1-68. Claims 38-65 and 68 are withdrawn from consideration. Claims 1-15, 17-25, and 66 are rejected under 35 U.S.C. 102(a) as being anticipated by Machin et al. (US 6,243,753) hereinafter *Machin* which incorporates by reference Shaw et al. (US 6,205,492) hereinafter *Shaw*. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over *Machin* and *Shaw* in view of Shilmover et al. ("Windows 2000 Power Toolkit") hereinafter *Shilmover*. Claims 26-37 and 67 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Machin* in view of Miloushev et al. (US 2003/0056205) hereinafter *Miloushev*.¹

By this paper, claims 1-17, 19-37, 66, and 67 have been amended.² Claim 18 has been cancelled. Accordingly, claims 1-17, 19-37, 66, and 67, are pending, of which claims 1, 26, 37, 66, and 67 are the only independent claims at issue.

The present invention is generally directed a run-time environment for network filter drivers. Claim 1, for example, recites a method for inserting a communication filter instance into a communication filter stack without disrupting the operation of one or more network protocol stacks so as to conserve processor and network resources. Claim 1 defines an abstract interface pausing operation of an operational communication filter stack to transition the operational communication filter stack to a paused communication filter stack without disrupting operation of one or more network protocol stacks. Next, claim defines the abstract interface transferring at least one data packet between a transport layer and data link driver of a protocol stack the abstract interface is situated within without performing any data filtering operations on the data packet subsequent to transitioning the operational communication filter stack to a paused communication filter stack and while the paused communication filter stack remains paused. Next, claim 1 defines the abstract interface causing a communication filter instance to be inserted into the paused communication filter stack while the one or more network protocol stacks continue to be capable of transferring data between corresponding transport and data link layers. Lastly, claim 1 defines the abstract interface restarting the paused communication filter stack to

¹ Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

² Support for the amendments to the claims are found throughout the specification, Figures, and previously presented claims, including, paragraphs [0047], [0052], [0057], [0072], [0079]-[0083], [0092], and [0094] and Figures 2A and 2C.

transition the paused communication filter stack back into the operational communication filter stack without disrupting operation of the one or more network protocol stacks.

Claim 26 is directed to a method claim similar to claim 1 for removing a communication filter instance from a communication filter stack. Claim 37 is a method claim similar to claims 1 and 26 including functional language. Claim 66 is directed to a corresponding computer program product for implementing the method of claim 1. Claim 67 is directed to a corresponding computer program product for implementing the method of claim 26.

Machin describes creating a raw data channel from an integrating component to a series of kernel mode filters. *Machin* describes allowing data to be accessible to kernel mode filters and other processing components while simultaneously allowing connection-oriented device drivers to be written in a simplified fashion. (Abstract). As depicted in Figures 21B, a filter graph 528 with one or more filters can be communicatively coupled to integrating component 502. (Col. 27, 1. 1 – Col. 28, 1. 4).

Shaw describes mechanism for interconnecting software drivers in kernel mode. *Machin* can utilize the interconnection mechanism of *Shaw* in the creation of filter graph 528. (*Machin* Col. 27, ll. 17-22). *Shaw* utilizes input and output pin instances to appropriately format data between filters. (Fig. 9B and Col. 22, ll. 16-22). Pin instances can have various states as described in Table 2. Among these states are a “stop” state, the initial state of the connection pin instance, “pause” state representing the mode of most resource usage and a correspondingly low processing latency to arrive at run state, and “run” state representing a mode where data is consumed.

Created filter graphs are reconfigurable to allow changes after initial creation. (Col. 20, ll. 38-39). When changes are made, state transitions need only occur on those connection pin instances that are in the stop state in order to assure correct stack depth parameter information. (Col. 20, ll. 39-42). Applicants submit that since the “stop” state is the initial state, added pin instances are transitioned to a “run” state, while other existing pin instances in a filter graph are not transitioned. Thus, other pin instances in a “run” state are not stopped. Accordingly, the filter graph is not paused, but instead new pin instances are activated within the filter graph. Further, the “stop” state being an initial state infers that a pin instance can not transition back into this state.

Machin and *Shaw* are silent with respect to any impact adding/removing a pin instance (e.g., input pin instance 526 in *Machin*) has on communication paths that utilize a corresponding filter graph (e.g., a communication path between TCP/IP and connection-oriented hardware that can utilize filter graph).

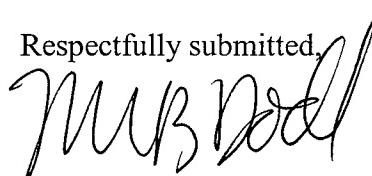
Accordingly, the art of record either singly or in combination fails to teach or suggest an abstract interface pausing operation of an operational communication filter stack to transition the operational communication filter stack to a paused communication filter stack without disrupting operation of the one or more network protocol stacks, as recited in claim 1. The art of record either singly or in combination also fails to teach or suggest the abstract interface transferring at least one data packet between a transport layer and data link driver of a protocol stack the abstract interface is situated within without performing any data filtering operations on the data packet subsequent to transitioning the operational communication filter stack to a paused communication filter stack and while the paused communication filter stack remains paused, as recited in claim 1. At least for either of these reasons applicants submit that claim 1 patentably defines over the art of record. For at least either of the same reasons, applicants submit that claims 26, 37, 66, and 67 also patentably define over the art of record.

In view of the foregoing, Applicants respectfully submit that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicants acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicants reserve the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicants specifically request that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 18th day of September, 2006.

Respectfully submitted,



RICK D. NYDEGGER
Registration No. 28,651
MICHAEL B. DODD
Registration No. 46,437
Attorneys for Applicant
Customer No. 47973

MBD:gcd
GD0000000069V001